No.

Date. / /

1. 1. F

2. T

3. F

4. F

2. A C C

No.
Date. / /

3(1) Index Tag1 Tag2

0b00

0b01

0b10

0b11

memory address 10 bits

set_index 2 bits ∴ set-associative

offset_size 4 bits and associativity = 2

way 2

tag 4 bits

(2)

Tag	Set-index	Blockoffset
4	2	4

(3) cache size = number of sets * Associativity * block_size

= 4 X 2 X 16

= 128 B

∴ this cache can contain 128 bytes of data

(4) Address

0b|0011|0100|0000	B
0b|0011|0110|1011	A
0b|1101|0001|0000	B
0b|0011|0011|1100	B
0b|0110|0001|0000	B
0b|0010|1111|1100	B
0b|1110|0010|0000	B
0b|1011|0000|0000	B
0b|1111|1111|1111	B
0b|0011|0010|1011	A.

Wengu

No.
Date. / /

4. (1) Total number of accesses to the cache

① repeat-time = 1, step-size = 2, array-size = 64.

$1 \times \frac{64}{2} = 32$

② array[i] = array[i] + 2333
write read 2 times one step

$\therefore 32 \times 2 = 64$

$\therefore$ the final answer = 64.

(2) Block size $16 = 2^4$ 4 bits

Num of sets $4 = 2^2$ 2 bits

Cache size 128

$\therefore$ Associativity $= \frac{128}{16 \times 4} = 2$

memory address 32 bits

...	00	0000
tag	index	Block offset

step-size = 2

$\therefore$ every step address + 2×4 = 8

...	00	0000
		+1000

$\therefore$ take the first two steps for example

(read) the first step we miss and the type of miss is compulsory miss (write) and then we will hit

the second step address memory +8, but it didn't change index and tag, just changes offset $\therefore$ for reading and writing, we have two hits.

Therefor the hit_rate $= \frac{3}{4} = 0.75$

(3) compulsory miss

(4) · 0.75

That's because we just have 8 blocks, and we need to get 16 different datas from memory, ~~that of which~~ whose index or tags are different.

So the data in the cache will be replaced, and then when 2nd repeat ~~cache~~, we still have a process which is similar with first repeat ~~cache~~ $\therefore$ the answer is 0.75 forever.

REMEMBER MEMORY

No.
Date. / /

(5) line 6 and line 7

the max hit rate is 0.875

for (int i=0 , i< array_size ; i += step_size)

for (int r=0; r<repeat_times; r++)

this is equal to the process that we perform two consecutive operations

so for the first step.

we hit 3 times and miss 1

for the second step

we hit 4 times . the reason is similar with (4)

So the answer is $0.875 = \frac{7}{8}$

(6) the hit rate converge to 1

hit rate = $\frac{2 \text{repeat\_times} - 1}{4 \text{repeat\_times}}$ ∴ converge to 1